Portfolio 5 - due 17th March 2020

Morten Street, Gustav Helmet, Anders Wheelman and Sigrid Snapfield

1. Checking input using R

Portfolio 5

```
pacman::p_load(tidyverse, reshape2, pastecs, WRS2)

data <- read.delim("alignment.txt", header = F, sep = "")

#creating the variables

duration_story1 <- c(35, 27, 27, 36, 26, 16, 29, 42, 33, 54, 22, 38, 43, 43, 21)


duration_story2 <- c(55, 33, 23, 37, 16, 30, 48, 40, 26, 24, 46, 27, 25, 27, 30)


ratings_story1 <- c(7,4,4,5,3,1,6,2,1,4,2,3,2,4,1)


ratings_story2 <- c(8,2,3,2,5,7,7,3,3,2,3,3,2,4,3)


# creating two different data frames and melting them for future t-tests


df1 <- data.frame(duration_story1, duration_story2)


df1_melt <- melt(df1)

## No id variables; using all as measure variables

df1_melt$variable <- as.factor(df1_melt$variable)


df2 <- data.frame(ratings_story1, ratings_story2)
```

```r
df2_melt <- melt(df2)

## No id variables; using all as measure variables

df2_melt$variable <- as.factor(df2_melt$variable)
```

1.a There was a significant difference between the durations of the two story types.

In order to decide which t-test to conduct the assumption of normality must be checked. If the data are normally distributed, we will conduct a parametric t-test and if not, we'll conduct a non-parametric t-test.

```r
round(pastecs::stat.desc(cbind(df1$duration_story1, df1$duration_story2), basic = F, norm = T), digits = 2)

##                  V1     V2

## median        33.00  30.00

## mean          32.80  32.47

## SE.mean        2.63   2.77

## CI.mean.0.95   5.64   5.94

## var          103.60 115.12

## std.dev       10.18  10.73

## coef.var       0.31   0.33

## skewness       0.27   0.61

## skew.2SE       0.23   0.52

## kurtosis      -0.83  -0.75

## kurt.2SE      -0.37  -0.34

## normtest.W     0.98   0.93

## normtest.p     0.94   0.31

# Both variables are not significantly different from a normal distribution according to Shapiro Wilk's test
```

```r
# looking at the data by making qq-plots

qqplot_dura_story1 <- ggplot(df1, aes(sample = duration_story1))

qqplot_dura_story1 + stat_qq() + stat_qq_line(colour = "red") + ggtitle("Duration of story 1 qq-plot")

qqplot_dura_story2 <- ggplot(df1, aes(sample = duration_story2))

qqplot_dura_story2 + stat_qq() + stat_qq_line(colour = "red") + ggtitle("Duration of story 2 qq-plot")

# The qq-plots don't look too shabby. We will conclude that the data are normally distributed and we'll continue with conducting a parametric t-test


# parametric t-test


t.test(value ~ variable, df1_melt)

##

##  Welch Two Sample t-test

##

## data:  value by variable

## t = 0.087292, df = 27.922, p-value = 0.9311

## alternative hypothesis: true difference in means is not equal to 0

## 95 percent confidence interval:

##  -7.489661  8.156327

## sample estimates:

## mean in group duration_story1 mean in group duration_story2

##                      32.80000                      32.46667
```

When conducting a parametric t-test, there was found no significant difference between the duration of the two stories, $t(27.92) = 0.087$, $p > .05$ ($p = .93$).

1.b Hypothesis: There was a significant difference between the ratings of the two story types.

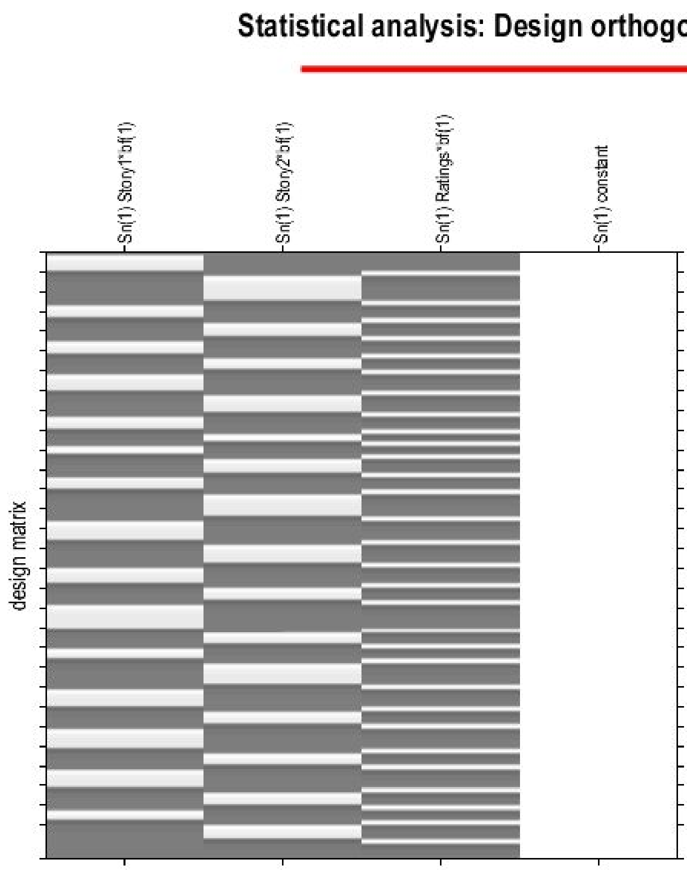
```
#checking normality

round(pastecs::stat.desc(cbind(df2$ratings_story1, df2$ratings_story2), basic = F, norm = T), digits = 2)

##               V1    V2

## median       3.00  3.00

## mean         3.27  3.80

## SE.mean      0.47  0.52

## CI.mean.0.95 1.01  1.11

## var          3.35  4.03

## std.dev      1.83  2.01

## coef.var     0.56  0.53

## skewness     0.41  0.95

## skew.2SE     0.36  0.82

## kurtosis    -0.94 -0.67

## kurt.2SE    -0.42 -0.30

## normtest.W   0.93  0.79

## normtest.p   0.27  0.00

# The ratings of story 1 could be normally distributed according to Shapiro Wilk's test, but since the data from the ratings for story 2 are significantly different from a normal distribution, we'll conclude that the data probably are not normally distributed.


# Visual inspection with qq-plots

qqplot_rat_story1 <- ggplot(df2, aes(sample = ratings_story1))

qqplot_rat_story1 + stat_qq() + stat_qq_line(colour = "red") + ggtitle("Ratings for story 1 qq-plot")

qqplot_rat_story2 <- ggplot(df2, aes(sample = ratings_story2))
```

```r
qqplot_rat_story2 + stat_qq() + stat_qq_line(colour = "red") + ggtitle("Ratings for story 2 qq-plot")

# Data cannot be assumed to be normally distributed and we will therefore conduct a non-parametric t-test.

# conducting non-parametric t-test

WRS2::yuen(value ~ variable, df2_melt)

## Call:

## WRS2::yuen(formula = value ~ variable, data = df2_melt)

##

## Test statistic: 0.166 (df = 15.39), p-value = 0.87034

##

## Trimmed mean difference:  -0.11111

## 95 percent confidence interval:

## -1.5348     1.3126

##

## Explanatory measure of effect size: 0.04
```

When conducting a non-parametric t-test there was found no significant difference between the ratings of the two stories, $t(15.39) = 0.17$, $p > .05$ ($p = .87$).

2. Create the model in SPM

2.a Make a screenshot and report the design matrix figure generated by SPM. How many columns does it have? What do the different columns represent?

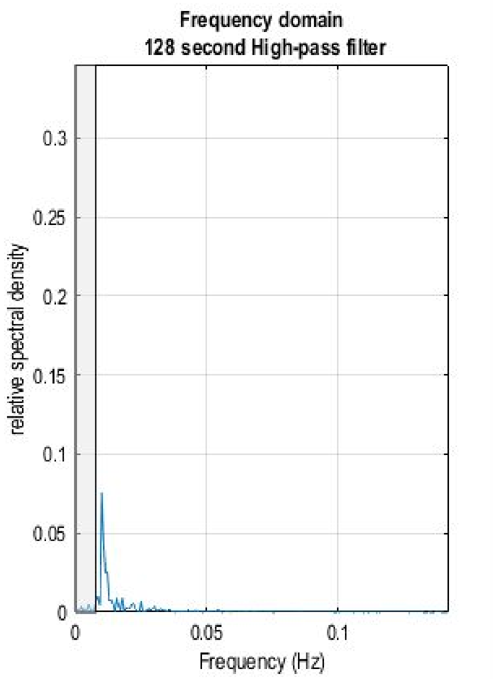


The columns represent our conditions. First two columns indicates when story 1 and story 2 are presented to the participant. Third column indicates when the participant rated the stories, explaining why it has 30 white intervals. Fourth column is a constant vector used to calculate an intercept.

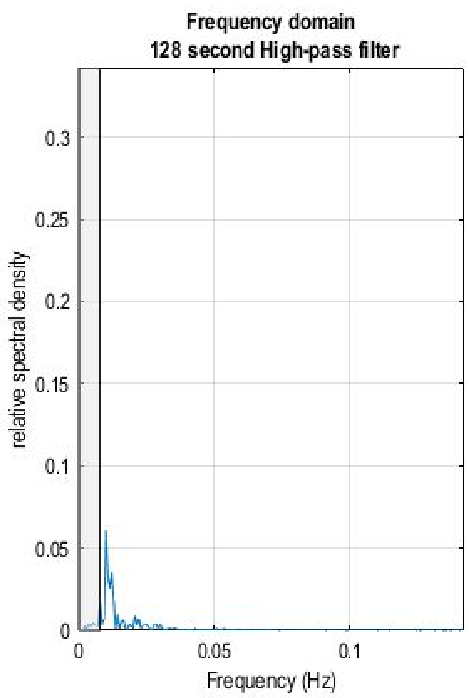
3. Checking the model

3.a Report periodogram plots of the Frequency domain for the three conditions and **3.b**.Eye-balling task: What are the most predominant frequencies for the three condition, as seen from these plots?

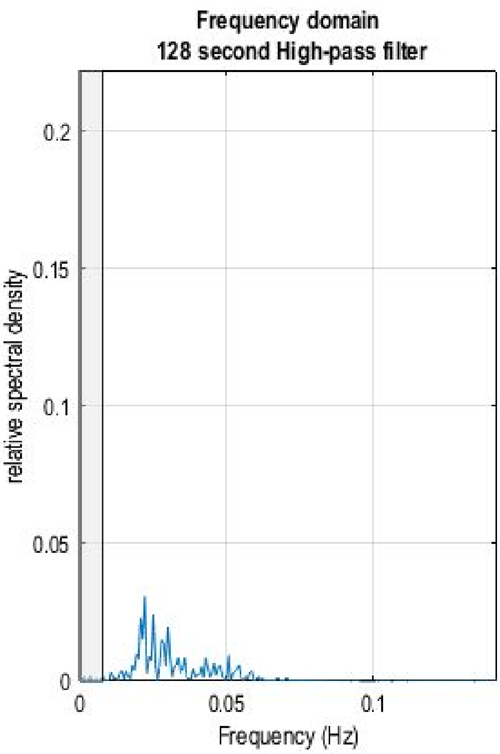
Story 1: Most dominant frequency is approx 0.01. All frequencies in the grey area are cut off due to our high pass filter.



Story 2: Again, most dominant frequency is approx 0.001. Again frequencies within the grey area are cut off.

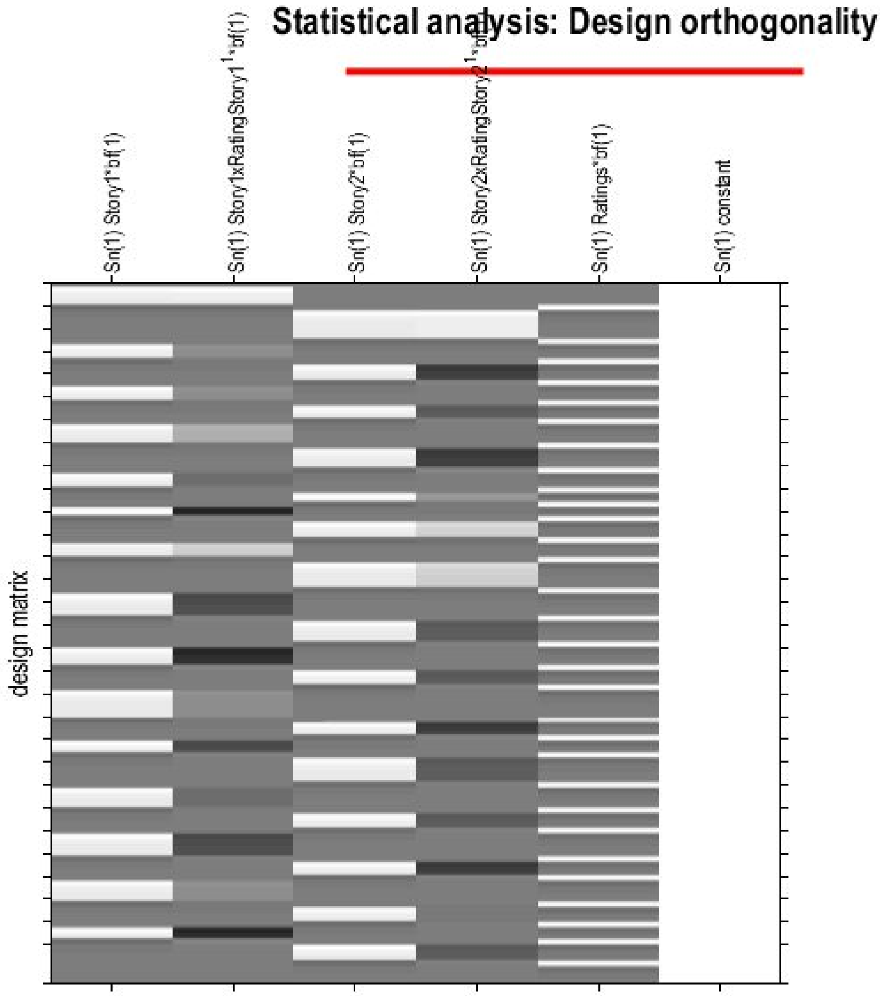


Ratings: The most dominant frequency is approximately 0.025. It is more widely distributed as it covers more frequencies. This predictor is not very influenced by the filter since the frequencies are not in the grey area.



4. Adding covariates

4.a Make a screenshot and report the new design matrix figure. How many columns does it now have? Which columns model the rating effects?

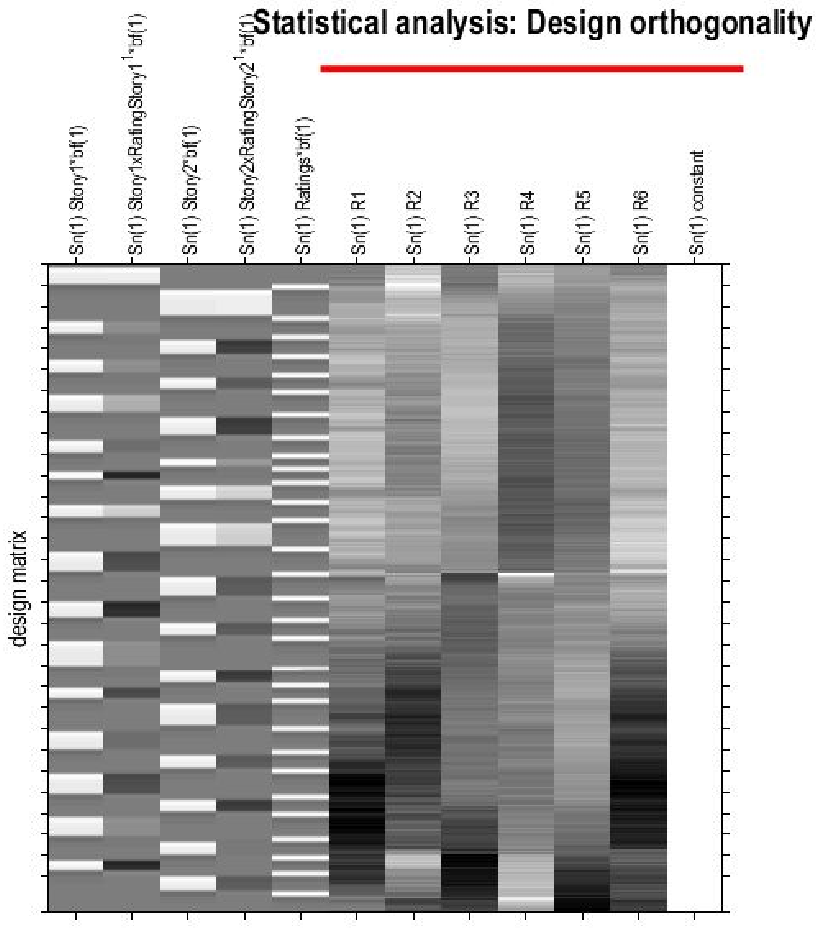


Two extra columns have been added to our design matrix. The column for “rating of story 1” is at [,2] and the column for “rating of story” is at [,4], representing the two covariates.

4.b Why is it important to subtract the mean?

Mean-centering the covariate makes interpretation of the effects easier, as the effects can now be interpreted as above or below the mean instead of some arbitrary intercept. In that way the covariate can be compared to its predictor.

4.c Make a screenshot and report the new design matrix figure. How many columns does it now have? Which columns are modeling the motion?

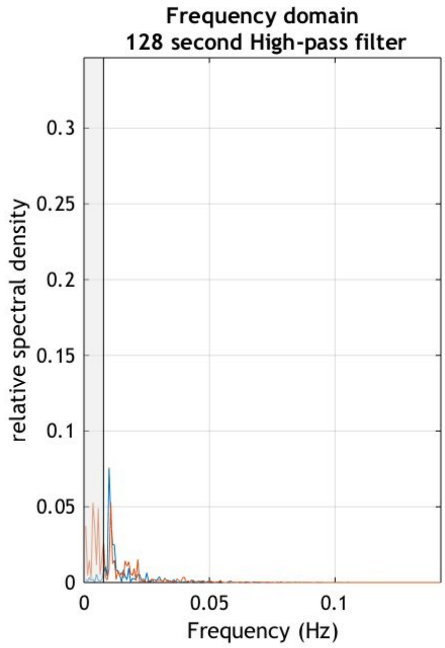


Now six columns have been added for the motion parameters. Now the model has 12 columns where [,6:11] models the motion parameters.

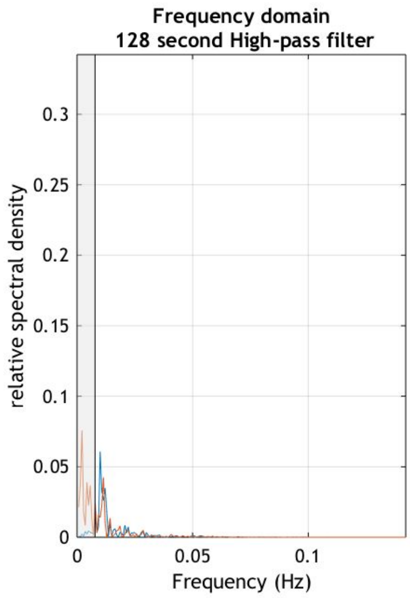
5. Checking the new model

5.a Report plots of the Frequency domain for the three conditions.
5.b Eye-balling task: What are the most predominant frequencies for the covariates, as seen from these plots?

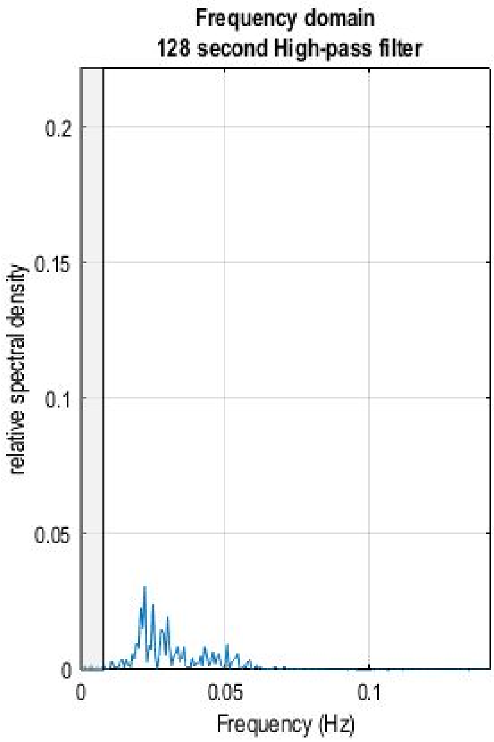
Story 1: Blue line is similar to before adding the extra covariate. So the red line represent the added rating and motion covariates. The dominating frequency for the red line is in the high-pass filter zone.



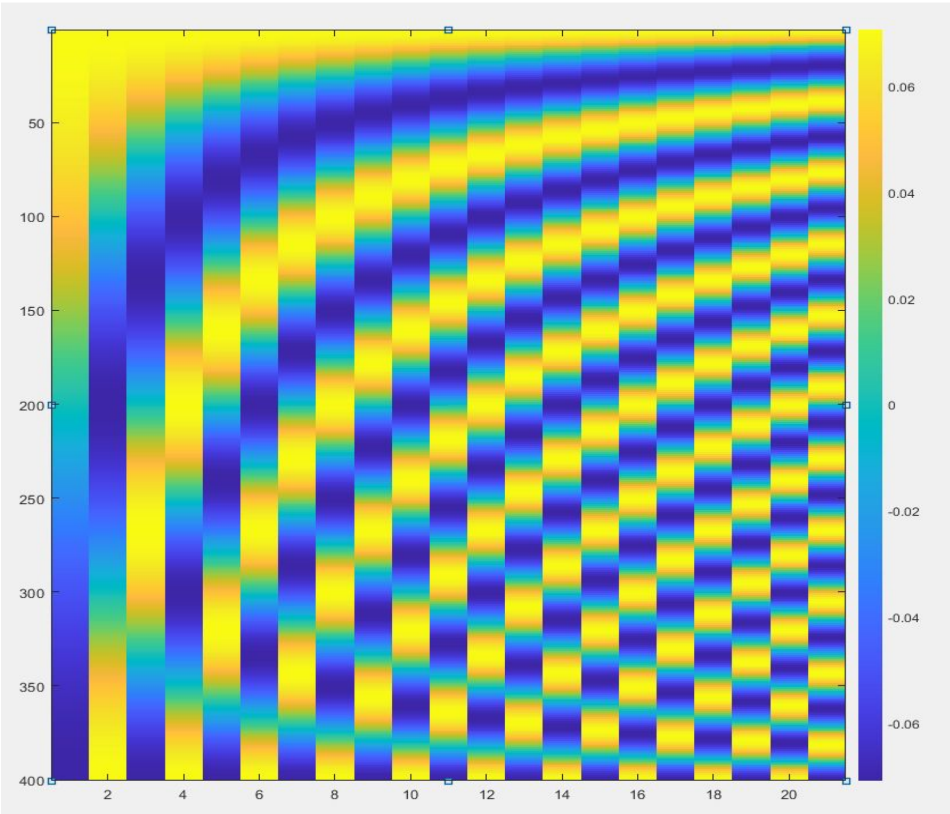
Story 2: Again, the added covariates for rating and motion do not pass through the filter. Their dominating frequencies are too low to pass through.

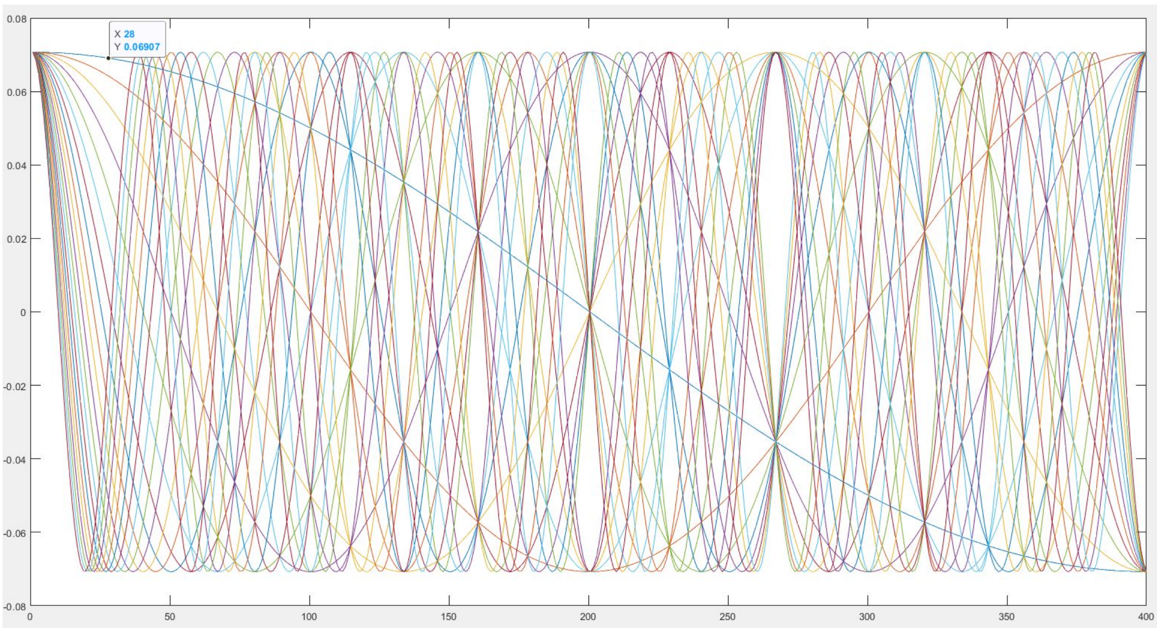


Rating: Similar to earlier frequency domain plots because we did not add any parametric modulations to the rating condition in the batch.



5.c The high-pass filter consists of low-frequency cosine-waves, which together can model any fluctuation below the specified frequency. Plot and report figures of the high-pass filter using these two lines in MatLab (you need to have loaded the SPM.mat file)





See 5.d for report on the graphs.

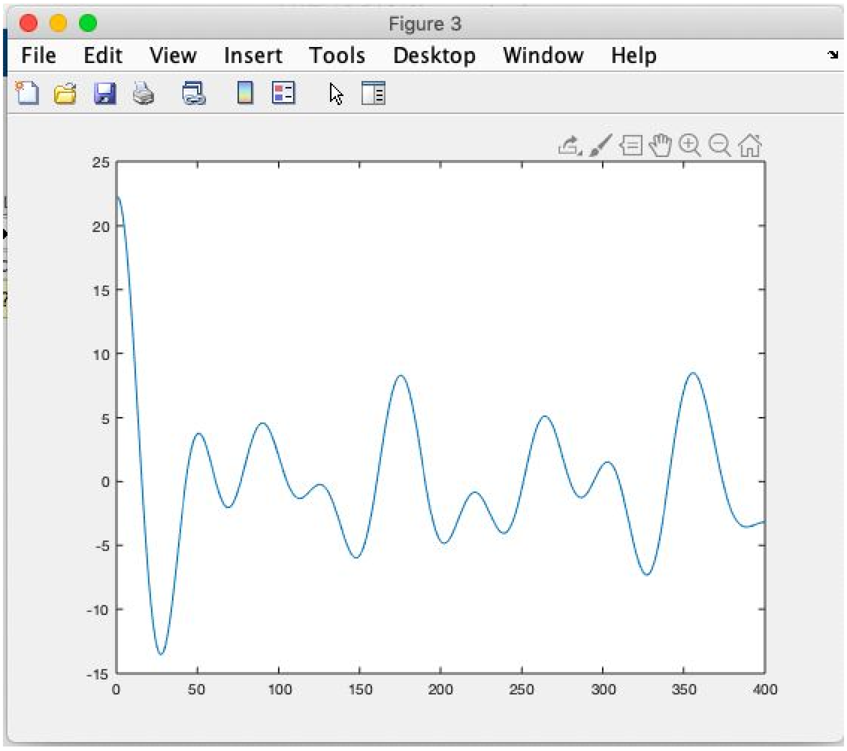
5.d How many cosine waves are in this specific high-pass filter?

The high pass filter filters out the 21 cosine waves that are below the chosen high pass filter of 128 seconds per wave. The 21 cosine waves are pictured in the plot above.

The whole scanning session lasts 1400 seconds (400 scans * 3.5 seconds between each scan). The plot above pictures the 21 possible cosine waves that are below the threshold of 128 seconds.
When dividing 1400 with 128, we get a threshold of 10.94 completed waves pr. session. Every cosine wave have half a wave more than the last one, starting at half a wave, which is why we end up with 21 different wave types that are filtered out since the threshold is at 21.88 “half waves” per session.

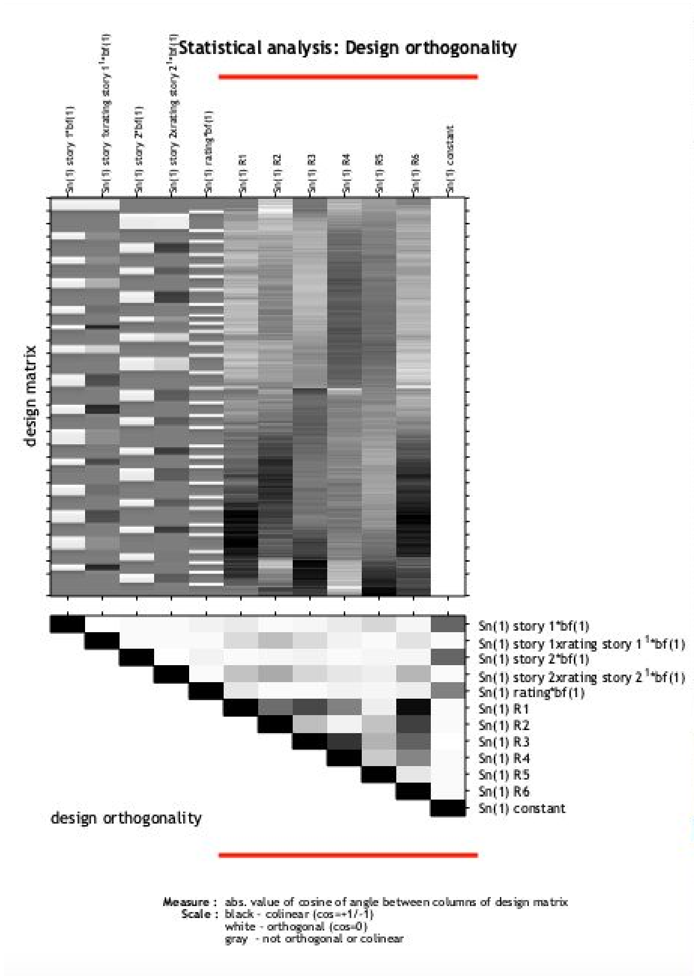
To get the frequency per second (Hz) we can dividing 10.94 by 1400. We then get 0.0078 Hz (waves per second). Everything below this frequency are filtered out by the high pass filter. This frequency threshold of 0.0078 is illustrated by the grey area seen in the frequency domain plots.

5.e - a hypothetical slow wave



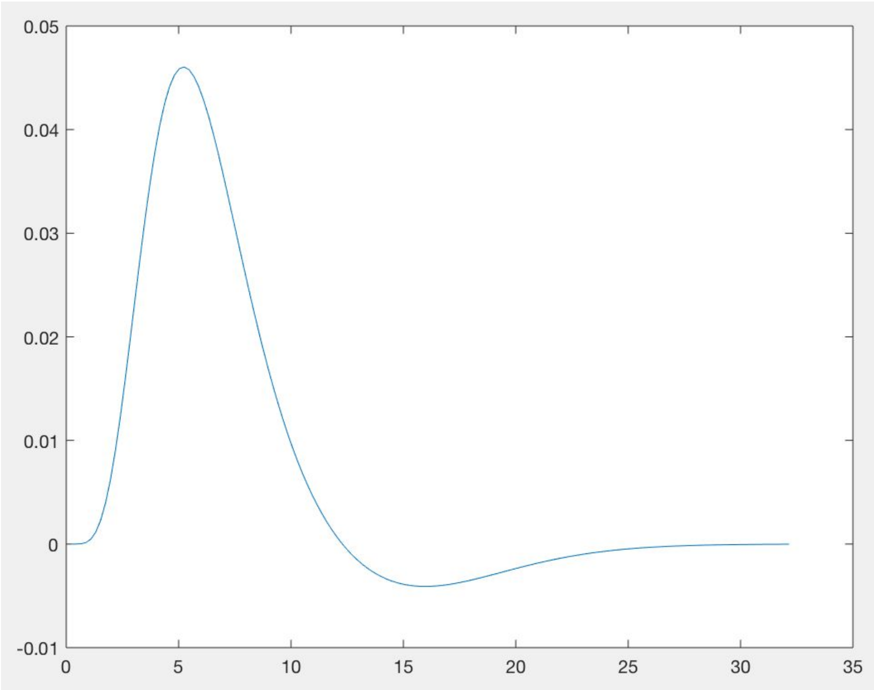
This hypothetical slow wave has a frequency of approximately 8 waves per session, which would be filtered out by the high pass filter if found in the data by the cosine wave number 16 from the plot in 5.c.

5.f Explore "design orthogonality". Which covariates are most correlated in the current design?



The design orthogonality matrix shows correlation between the covariates included in the model. The plot shows a strong correlation between R1 (x-coordinate) and R6 (yaw). In general the motion parameter R6 is strongly correlated with the other motion parameters. These correlations seem plausible as head movements rarely happens in one dimension.

5.g Plot and report the hemodynamic response function (HRF)



The hemodynamic response has a delay of 5-6 seconds before the level of oxygenated blood in the voxel drops. We see a time interval where the BOLD-signal is lower than initially before it “stabilizes”.